NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A HIGH POWERED RADAR INTERFERENCE MITIGATION TECHNIQUE FOR COMMUNICATIONS SIGNAL RECOVERY WITH FPGA IMPLEMENTATION

by

Geoffrey R. Meager

March 2017

Thesis Advisor: Ric A. Romero
Co-Advisor: Zachary Staples

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704–0188*

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** March 2017	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
A HIGH POWERED RADAR INTERFERENCE MITIGATION TECHNIQUE FOR COMMUNICATIONS SIGNAL RECOVERY WITH FPGA IMPLEMENTATION

5. FUNDING NUMBERS
N/A

6. AUTHOR(S) Geoffrey R. Meager

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER N/A

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
N/A

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

In this thesis, we investigate the demodulation of a communications signal that is interfered with by a high-powered radar signal. We choose quaternary phase-shift keying (QPSK) modulation for illustration. The radar phase offset is initially assumed to be fixed. Later, a practical phase sequence is added to the phase offset. A least-squares estimator (LSE) is used to estimate the amplitude and fixed-phase offset of the interfering radar signal to be used for interference cancellation. Then, we utilize a maximum-likelihood detection (MLD) receiver. We show that the QPSK symbol error ratio (SER) improvement in the radar interference depends on collection time. The variance of the estimate increases as the QPSK signal-to-noise ratio (SNR) increases. The increase in variance affects SER. SER results approach the ideal with increasing collection times. For the case where the phase offset includes a phase sequence, the original LSE technique is modified. Later, a system is created in Simulink and converted to hardware-description language (HDL). The SER performance of the HDL implementation in hardware is compared to those of the signal model. SER performance is somewhat degraded in the hardware implementation.

[Reprinted in part, with permission, from G. Meager, R.A. Romero, Z. Staples, "Estimation and cancellation of high powered radar interference for communication signal collection," in *2016 IEEE Radar Conf.*, 2016, pp. 1–4, doi: 10.1109/RADAR.2016.7485263.]

14. SUBJECT TERMS parameter estimation; matched-filter detection; QPSK; radar; interference; LSE, cyber, electronic warfare	**15. NUMBER OF PAGES** 69
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

A HIGH POWERED RADAR INTERFERENCE MITIGATION TECHNIQUE FOR COMMUNICATIONS SIGNAL RECOVERY WITH FPGA IMPLEMENTATION

Geoffrey R. Meager
Civilian, Raytheon (SAS) Corporation
B.S.E., Walla Walla College, 2002

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2017

Approved by: Dr. Ric A. Romero
Thesis Advisor

CDR Zachary Staples
Co-Advisor

Dr. R. Clark Robertson
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In this thesis, we investigate the demodulation of a communications signal that is interfered with by a high-powered radar signal. We choose quaternary phase-shift keying (QPSK) modulation for illustration. The radar phase offset is initially assumed to be fixed. Later, a practical phase sequence is added to the phase offset. A least-squares estimator (LSE) is used to estimate the amplitude and fixed-phase offset of the interfering radar signal to be used for interference cancellation. Then, we utilize a maximum-likelihood detection (MLD) receiver. We show that the QPSK symbol error ratio (SER) improvement in the radar interference depends on collection time. The variance of the estimate increases as the QPSK signal-to-noise ratio (SNR) increases. The increase in variance affects SER. SER results approach the ideal with increasing collection times. For the case where the phase offset includes a phase sequence, the original LSE technique is modified. Later, a system is created in Simulink and converted to hardware-description language (HDL). The SER performance of the HDL implementation in hardware is compared to those of the signal model. SER performance is somewhat degraded in the hardware implementation.

[Reprinted in part, with permission, from G. Meager, R.A. Romero, Z. Staples, “Estimation and cancellation of high powered radar interference for communication signal collection,” in *2016 IEEE Radar Conf.*, 2016, pp. 1–4, doi: 10.1109/RADAR. 2016.7485263.]

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ADC	Analog-to-Digital Converter
AWGN	Additive White Gaussian Noise
CONOP	Concept of Operation
C-SNR	Communications Signal-to-Noise Ratio
CW	Continuous Wave
DAC	Digital-to-Analog Converter
EM	Electromagnetic
FPGA	Field-Programmable Gate Array
GUI	Graphical User Interface
HDL	Hardware Description Language
LSE	Least-Squares Estimation
MLD	Maximum-Likelihood Detection
QPSK	Quaternary Phase-Shift Keying
RCR	Radar-to-Communications Ratio
R-SNR	Radar-to-Noise Ratio
SER	Symbol Error Rate
SIGINT	Signals Intelligence
VHDL	Verilog Hardware Description Language

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank my advisors, Dr. Ric Romero and CDR Zachary Staples, for their indelible mark on my education and for their patience, guidance and approachability throughout this process.

I would also like to thank my beautiful wife, Sarah, and my two young children for their enduring understanding of my absence at times as a husband and father. Lastly, I thank my Lord and Savior Jesus Christ. Without His grace, this would all be meaningless.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

This section was previously published, in part, by IEEE in [1].[1]

The utilization of the electromagnetic (EM) spectrum for communications and sensing applications today is ubiquitous in both the commercial and military domains. Traditionally, radar and communications systems are managed through frequency (spectrum) separation. Today's EM environments increasingly do not allow for this luxury [2]. This dictates a need to manage the use of the EM spectrum efficiently such that all required communications and sensing functions can coexist and yet still operate as designed [3]–[6]. There may be times when it is necessary to have communications and radar signals operating in the same band or even on the same carrier frequency [3].

Obvious applications include detecting communications signals embedded in radar signals or other forms of interference, a.k.a. signals intelligence (SIGINT) [7]. This can also be considered a noise-based communications jamming application [8]. Usually, the radar is not in the same frequency band as the communications signal, but the high-powered nature of some radar systems disrupts communications nonetheless. Also, this can be looked at as the communications signal interfering with the radar [5], [9], [10]. Lastly, a very interesting and common problem is that of a high-powered radar signal interfering with a communications system [1], [11]. This situation is depicted in Figure 1. The signal S_2 is the communications signal we are interested in demodulating, where r is the radar interference, q is the communications signal and w is noise.

[1] Reprinted, with permission, from G. Meager, R.A. Romero, Z. Staples, "Estimation and cancellation of high powered radar interference for communication signal collection," in *2016 IEEE Radar Conf.*, 2016, pp. 1–4. This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States. IEEE will claim and protect its copyright in international jurisdictions where permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

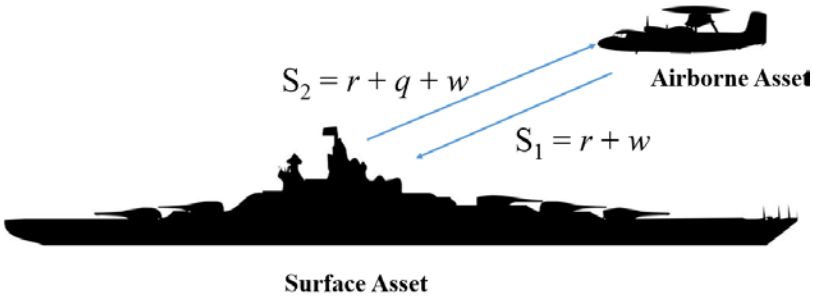

Figure 1. One-Way Communications System with a Receiver onboard an Airborne Asset Being Interfered with by a Radar Signal. Adapted from [12], [13].

The two situations we consider are high-power-in-band radar interference and the common-carrier concept of operations (CONOP). The emphasis of this thesis is at the baseband signal processing level; thus, the estimation, cancellation and demodulation algorithms of the communications signal at the receiver are implemented in baseband.

A. OBJECTIVE AND APPROACH

In this thesis, we consider a communications system's receiver being interfered with by a high-powered radar in the same carrier band. Our goal is to estimate and subsequently cancel the radar interference such that we can demodulate the communications signal. Our first step in interference cancellation is to estimate the parameters of the radar signal, namely the phase and amplitude. In our work, we consider a pulsed-Doppler waveform, where we try to demodulate the parts of the communications signal that coincide in time with the radar pulses of the interfering radar signal at the communications system receiver. We also consider a continuous–wave (CW) radar signal where the radar interferes with a substantial or even whole portion of the communications signal spectrum. For the purposes of illustration, we employ quaternary phase-shift keyed (QPSK) modulation for our communications signal. Initially in the case of CW radar, we assume that the radar phase remains the same from one QPSK symbol to the next. In the case of the pulsed-Doppler radar, we also assume as much. In other words, the radar phase from pulse to pulse remains the same, where each pulse duration is the same as the QPSK symbol duration.

In practice, the radar phase and amplitude are usually unknown at the QPSK receiver. As such, we do not assume knowledge of these parameters; hence, we attempt to estimate both parameters in order to perform a version of interference cancellation. Once the radar signal is estimated, it is subtracted from the received signal. The resulting signal is routed through a maximum-likelihood detector (MLD), which is a bank of four filters matched to the four symbols of the QPSK constellation. To evaluate if we can demodulate the QPSK signal effectively after interference cancellation, we generate symbol error ratio (SER) curves as a function of communications signal-to-noise ratio (C-SNR). We specifically use C-SNR to differentiate it from radar signal-to-noise ratio (R-SNR). Further evaluation is made by varying the radar-to-communications ratio (RCR).

Another objective of this thesis is to produce a prototype hardware design for proof of concept to recover communications symbols from radar in-band interference. Symbol error ratio plots (SERs) are used to evaluate the proof of concept. The SER results demonstrate it is possible to estimate and cancel the radar energy in order to demodulate the communications signal. In applications where a large number of samples can be acquired, our interference mitigation performance approaches the ideal QPSK SER curve.

We begin this thesis with the one-way link CONOP, aka S_2 in Figure 1. The ground asset in Figure 1 has a communications and radar transmitter, and the airborne asset has a communications receiver. The main beam of the ground-based radar is assumed to be pointing line-of-sight (LOS) at the receiving aperture of the airborne asset.

B. THESIS OUTLINE

A signal model implemented in MATLAB is developed in Chapter II using signal processing, radar and communications systems theory. A number of SER plots are produced to distinguish the effects of varying certain design parameters, such as number of samples used in the estimation process. These SER plots demonstrate the relative importance of each design consideration through the estimation, cancellation and demodulation process.

In Chapter III, an equivalent Simulink implementation is built to utilize the capability inherent in Fixed-Point Designer to produce hardware description language (HDL) code for rapid prototyping. Simulink is a model-based design tool within the MATLAB suite. Fixed-Point Designer is a toolkit that can be used to convert MATLAB and Simulink designs to HDL. This HDL code is used to program a field-programmable gate array (FPGA). During various steps of this process, we compare results to the signal model from Chapter II. The hardware results from Chapter III are compared with the signal model in Chapter II. The hardware results are shown to approach those found in the MATLAB results.

In Chapter IV, results from the preceding two chapters are summarized, discussed and compared. Follow-on work is also discussed.

II. SIGNAL MODEL

We develop an interference mitigation algorithm to demodulate a communications signal that is interfered by a high-powered radar operating on the same frequency carrier. Monte Carlo (MC) statistical results are presented in the form of SER plots.

Further, a corollary objective is to understand the relationship between the variance of the interference estimate and the SERs after interference cancellation and demodulation of the communications signal; thus, variance is derived and calculated for each C-SNR and is included in some resulting SER plots.

This chapter was previously published, in part, by IEEE in [1].

A. SIGNAL DEFINITIONS

We assume a continuous-time, complex-valued baseband model such that the signal received at the communications receiver is simply $\tilde{y}(t) = \tilde{r}(t) + \tilde{q}(t) + \tilde{w}(t)$ with $\tilde{r}(t)$ being the complex-value radar interference radar signal, $\tilde{q}(t)$ being the complex-valued QPSK communications signal to be demodulated and $\tilde{w}(t)$ being complex additive white Gaussian noise (AWGN). Actual signal processing occurs after analog-to-digital (A/D) sampling. As such, we assume a discrete signal model where the Nyquist rate is followed [1]. It is also convenient to assume normalized sampling time T_s, where $T_s = 1$. In other words, the discrete model for the received signal is given by

$$\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{r}} + \tilde{\boldsymbol{q}} + \tilde{\boldsymbol{w}} . \tag{1}$$

We define the radar signal to have amplitude A, initially with some constant phase offset defined by distance between the transmitter and receiver. Again, we assume the radar to be CW or, equivalently, a pulsed-Doppler radar system in which we ignore the quiescent periods since the interference signal is not present during those periods. We let θ_r be the unknown, fixed-phase offset; thus, the complex-valued radar baseband signal is given by

$$\tilde{r} = Ae^{j\theta_r}. \tag{2}$$

The communications signal is defined as a complex QPSK signal with amplitude *Q* and phase $\boldsymbol{\varphi_q}$, which is given by

$$\tilde{\boldsymbol{q}} = Qe^{j\boldsymbol{\varphi_q}}. \tag{3}$$

The symbols are a random draw from the set

$$\varphi_q \in \left[\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4}\right]. \tag{4}$$

The noise $\tilde{\mathbf{w}}$ is defined as complex, additive and normally distributed with sample variance σ^2 [1].

We initially assume the received radar phase is constant; however, this assumption is not realistic for the real world implementation we are considering. This is because in pulsed radar, the phase from pulse to pulse may be different despite being coherent. Also, for a coded CW radar, the phase offset is added to the phase sequence. We assume that the phase sequence is known in the receiver and cyclical for long radar waveforms. We assume the phase offset to remain unknown; thus, our phase-sequence radar signal is given by

$$\tilde{\boldsymbol{r}} = Ae^{j(\theta_r + \boldsymbol{\varphi_r})}, \tag{5}$$

where θ_r is the unknown-phase offset and $\boldsymbol{\varphi_r}$ is the known-phase sequence of the transmitter.

B. INTERFERENCE ESTIMATION

This section was previously published, in part, by IEEE in [1].

The first step in mitigation of the radar interference for the demodulation of QPSK symbols is the estimation of the interference signal magnitude and phase offset, where least-squares estimation (LSE) is used [1]. We define the radar signal to be $\tilde{\mathbf{r}} = \tilde{r}\mathbf{1}$. The notation $\mathbf{1}$ is used to represent an *N*-length vector of ones. The LSE of our interference radar signal thus takes the form

$$J(\hat{\tilde{\mathbf{r}}}) = \left\| \tilde{\mathbf{y}} - (\tilde{r}\mathbf{1} + \tilde{\mathbf{q}}) \right\|^2, \tag{6}$$

where $\hat{\boldsymbol{r}}$ is the resulting estimate and $\hat{\boldsymbol{r}} = \hat{r}\mathbf{1}$. The radar power is assumed to be much greater than the communications power, thus Equation (6) reduces to

$$J(\hat{\tilde{\mathbf{r}}}) \cong \left\| \tilde{\mathbf{y}} - (\tilde{r}\mathbf{1}) \right\|^2. \tag{7}$$

Equation (7) is equivalent to

$$J(\hat{\tilde{\mathbf{r}}}) \cong \left(\tilde{\mathbf{y}} - \tilde{r}\mathbf{1} \right)^H \left(\tilde{\mathbf{y}} - \tilde{r}\mathbf{1} \right) \tag{8}$$

where *H* is the operation corresponding to the conjugate-transpose. It can be shown with the help of [14] that the radar estimate is given by

$$\hat{\tilde{r}} = \frac{\sum_{n=0}^{N-1} \tilde{y}[n]}{\sum_{n=0}^{N-1} 1^2}, \tag{9}$$

which reduces to

$$\hat{\tilde{r}} = \frac{1}{N} \sum_{n=0}^{N-1} \tilde{y}[n], \tag{10}$$

where *n* is the unit-time index.

The estimation procedure is now established. This procedure is depicted in Figure 2. The first *N* samples of the received signal are summed and then divided by *N* as defined in Equation (10); thus, in the fixed-phase case, taking the mean of *N* received samples gives the interference estimate.

In the case of the interference having a phase sequence, the known sequence is added to the fixed-offset estimation; thus, we develop additional processing to estimate the radar interference.

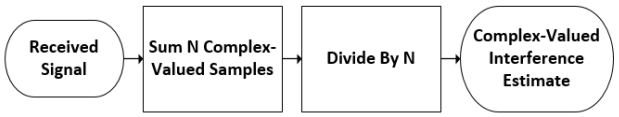

Figure 2. The Process of Estimating the Radar-Inference Magnitude and Phase Offset in the Constant-Phase Case

This additional processing is depicted in Figure 3. The fixed-phase offset due to the transmission distance still occurs. Additionally, as shown in Figure 3, the known-phase sequence is subtracted from the received signal.

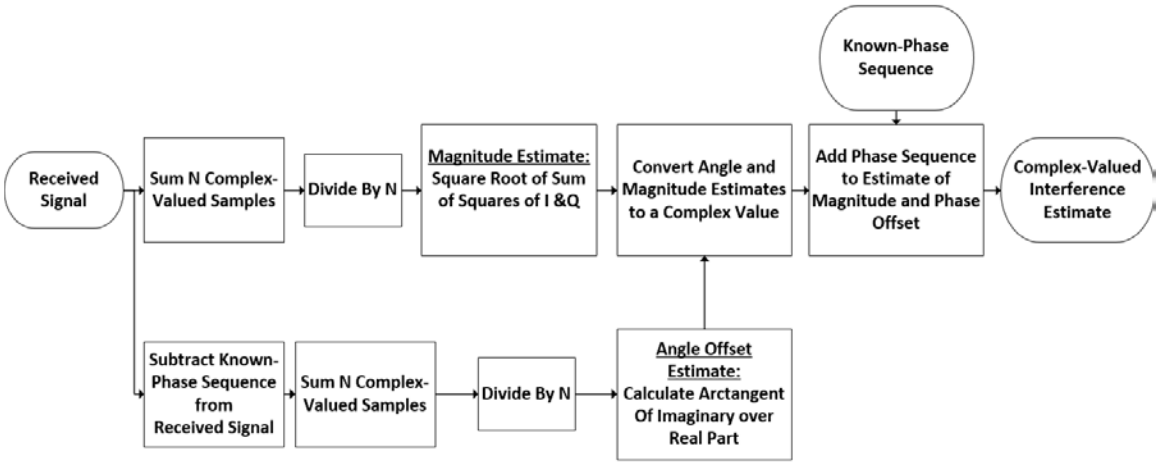

Figure 3. The Process of Combining the Known-Phase Sequence of the Interference with the Radar Magnitude and Fixed-Phase Offset Estimates to Produce a Complex-Valued Interference Estimate

Due to large R-SNR, we can easily detect the beginning of a radar signal against noise. Indeed, a simple energy detector suffices. In other words, we can easily align the known-phase sequence in the receiver with the received radar signal prior to subtraction. The known-phase sequence is then subtracted from the received signal as described in Figure 3 and depicted in Figure 4. The resulting difference between the measured and known-phase sequence is the phase offset for each given radar pulse or phase sub-code in the case of CW radar. Then, the LSE can be used to derive an estimate (to find the phase offset) described in Figure 3 and defined in Equation (10).

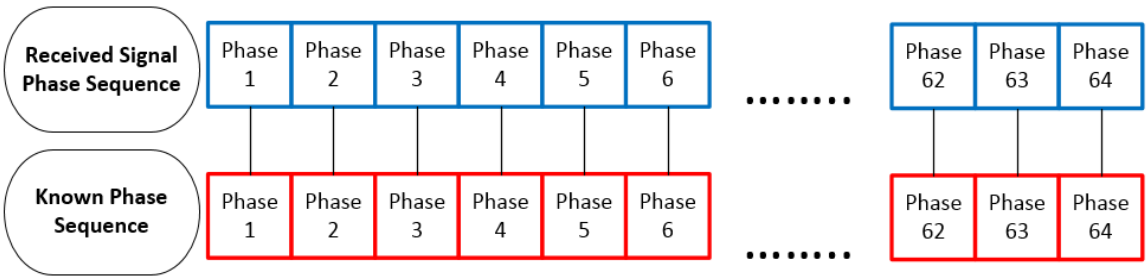

Figure 4. The Signal Corresponding to the Known-Phase Sequence Is Subtracted from the Received Signal to Obtain the Phase Offset of Each Sample

After the summation of N samples and division by N, the phase offset and magnitude are calculated. After the angle-offset and magnitude estimates are converted to a complex number, we add the complex-valued signal corresponding to the phase sequence to the estimate. This process is depicted in Figure 3.

It is readily apparent from Equation (10) that the estimated radar signal becomes more accurate as N increases; thus, the intent in any implementation is to gather as many samples N as possible. In other words, the longer the collection time, the better chance there is to accurately estimate the radar signal [1].

Unfortunately for the small data set case, the variance of the estimate worsens. Indeed, for this case, we derive and calculate the variance of the interference estimate as a function of the number of samples utilized N. Each sample period coincides with a full QPSK symbol period, a full radar-phase period and an equivalent period of noise. To present SER demodulation results for the communications signal, we show SER as a function of N such that we can quantify the effect of decreasing or increasing N. The relationship between radar estimation variance and communications SER versus C-SNR is also shown. It is demonstrated that radar estimation variance has an appreciable effect for very low collection times (N = 8, 16) and is mitigated effectively for practical collection times such as N equal to or greater than 64 [1].

The variance of the estimator given in Equation 10 plays a role in the SER of the QPSK receiver. We define the variance of our estimator as

$$\text{var}(\hat{\tilde{r}}) = \text{var}\left(\frac{1}{N}\sum_{n=0}^{N-1}\tilde{y}[n]\right) \tag{11}$$

Equation (11) is equivalent to

$$\text{var}(\hat{\tilde{r}}) = \frac{1}{N^2}\text{var}\left(\sum_{n=0}^{N-1}\tilde{y}[n]\right) \tag{12}$$

Further, Equation (12) can be reduced to

$$var(\hat{\tilde{r}}) = \frac{1}{N}var(\tilde{y}[n]). \tag{13}$$

The variance of the received signal $\tilde{y}$[n] is by definition

$$\text{var}(\tilde{y}[n]) = E\left[|\tilde{y}[n]|^2\right] - \left|E\left[\tilde{y}[n]\right]\right|^2 \tag{14}$$

The last term in Equation (14) is the magnitude squared of the expected value of the received signal. Recall that the radar phase sequence is known and that the phase offset and magnitude are fixed. Further, assuming that the communications signal and noise are independent, this can be broken down into its constituent parts as

$$|E[\tilde{y}[n]]|^2 = |E[\tilde{r}[n]]|^2 + |E[\tilde{q}[n]]|^2 + |E[\tilde{w}[n]]|^2 . \tag{15}$$

Again, the noise $\tilde{w}[n]$ is zero mean. Additionally, the communications signal is uniformly distributed across the complex plane with each sample having a probability of 1/4. Thus, the mean of the communications signal is also zero. It follows that the last two terms in Equation (15) are zero. Recalling that T_s = 1 and, with the assumption of the radar interference having a fixed phase, Equation (15) can be rewritten as

$$|E[\tilde{y}[n]]|^2 = |E[\tilde{r}[n]]|^2 . \tag{16}$$

Equation (16) is also true under the phase-sequence assumption. Recall from Equation (5) that the radar interference consists of a fixed-phase offset as well as a phase-sequence component.

The square of the magnitude of the expected value of the radar interference in Equation (16) can be shown to be

$$|E[\tilde{r}[n]]|^2 = E_r, \tag{17}$$

where E_r is the energy of the radar interference.

Substituting Equation (17) into Equation (14) yields

$$var(\tilde{y}[n]) = E[|\tilde{y}[n]|^2] - E_r. \tag{18}$$

Next, we derive the value of the expected value of the magnitude squared in terms of the received signal and its components:

$$E[|\tilde{y}[n]|^2] = E[|\tilde{r}[n]|^2 + |\tilde{q}[n]|^2 + |\tilde{w}[n]|^2]. \tag{19}$$

With the radar signal not being random and the noise and communications signal being independent, it can be shown that

$$E|\tilde{r}[n]|^2 = E_r, \tag{20}$$

$$E|\tilde{q}[n]|^2 = E_q, \tag{21}$$

and

$$E|\tilde{w}[n]|^2 = \sigma^2, \tag{22}$$

where E_q is the energy of the communications signal and σ^2 is the sample variance of the noise. Thus,

$$E[|\tilde{y}[n]|^2] = E_r + E_q + \sigma^2. \tag{23}$$

Substituting Equation (23) into Equation (18), we get

$$var(\tilde{y}[n]) = E_r + E_q + \sigma^2 - E_r, \tag{24}$$

which reduces to

$$var(\tilde{y}[n]) = E_q + \sigma^2. \tag{25}$$

Finally, we substitute Equation (25) into Equation (13) to get

$$var(\hat{r}) = \frac{(E_q + \sigma^2)}{N}, \tag{26}$$

giving an expression for the variance of the interference estimate in terms of the communications and noise signal energies and number of samples.

That the radar estimate variance increases as E_q increases while C-SNR increases is an interesting result [1]. Instead, only noise, number of samples (or symbols) taken and energy of the QPSK signal affect the variance of the radar estimate. Equation (26) suggests that the QPSK energy should be kept low in order to keep the variance of the radar estimate low, which is opposite what is conventional in communications theory; i.e., we should increase communications power to better SER or capacity. As a result, we strive to understand the trade-off between radar estimate variance, QPSK energy and their respective relative effects on SER performance.

C. DEMODULATION

This section was previously published, in part, by IEEE in [1].

The first step in the demodulation process is the subtraction of the interference estimate from the received signal. We refer to this process as the cancellation of the interference. This entails subtracting the output of Figure 2 (or 3) from the input, depending on the assumption. Theoretically, if the radar interference were estimated with a variance of 0 (which we know is not the practical case), the resulting signal after subtraction would simply be the QPSK signal plus the noise [1].

A bank of filters matched for each of the QPSK symbols is used to demodulate the signal after cancellation. The matched filters are defined as the complex conjugate of each QPSK symbol defined in Equation (4). For each sample, after the filter bank, the resulting symbol is compared with the original symbol. These comparisons provide for SER results utilizing Monte Carlo methods.

D. INTERFERENCE MITIGATION PROCEDURE

Using the signal definitions from Sections A through C of this chapter, we describe the procedure used to produce the SER plots. Following are the rest of the steps in the procedure, which we depict in Figure 5.

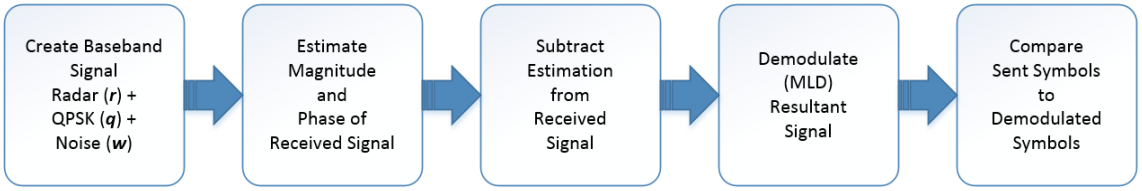

Figure 5. Estimation, Cancellation, Demodulation and Monte Carlo Procedures in MATLAB

We define and combine the radar, communications and noise baseband signals using MATLAB. This signal is assumed to be processed after the analog-to-digital converter (ADC) in the communications systems receiver. The second step is to use an LSE to estimate the phase offset and magnitude of the received signal. The assumption is that the radar phase and magnitude parameters are dominant in the received signal; thus, the estimations approach the interference radar parameters as the variance of the estimation process is lowered. Additionally, the phase sequence is combined with the results of the fixed-phase and magnitude LSE estimates, which yields the entirety of the interference estimate. Again, this process is depicted in Figure 3. This full interference estimate is subtracted from the received signal. The intent is that after the subtraction function, only the QPSK and noise remain. The fourth step in the procedure is to use a bank of four MLDs matched to each QPSK symbol to estimate which symbol was received. Each random symbol generated in step 1 is compared to each respective symbol demodulated symbol from step 5. This comparison was done 10^7 times utilizing the Monte Carlo (MC) method for each C-SNR. The RCR was fixed at 20 dB, and the C-SNR was varied from 0 to 15 dB.

E. RESULTS

1. Results for Fixed-Phase Case

We are interested both in the case where there is a significant data set (number of received signal measurements) and a small data set in the estimation process [1]. First, we consider the former where signal collection happens for a long duration of time (as in some signal collection applications where latency is not an issue). In this case, we show that complete demodulation of the communications signal is feasible (i.e., the radar signal

is mitigated). In other words, SER for the communications signal approaches the theoretical SER for QPSK demodulation. We consider the case of large collection times where we let $N = 10^7$ to produce asymptotic results. These results for the fixed-phase case are shown in Figure 6.

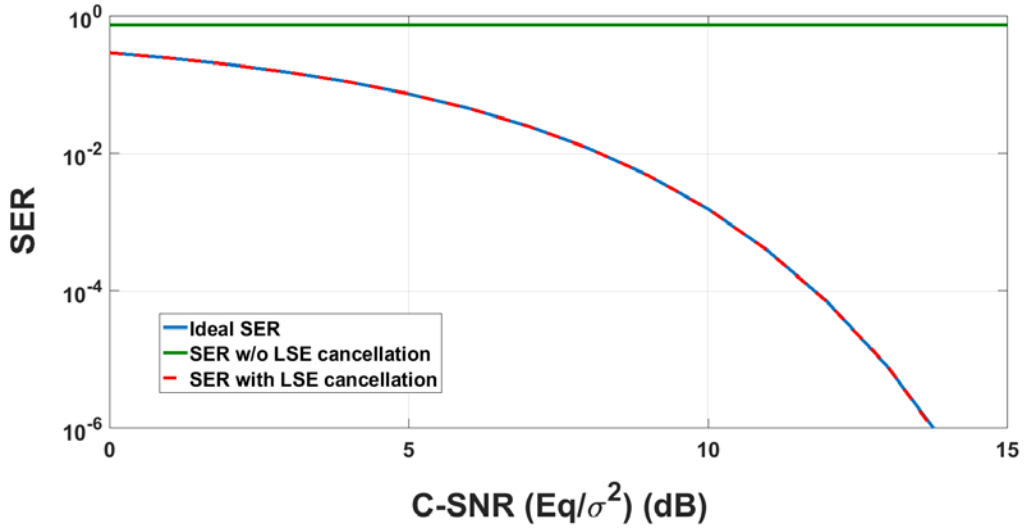

Figure 6. SER Performance for Large Collection Times. Adapted from [1].

The curve utilizing LSE cancellation (dashed red) approaches the ideal SER curve. The SER without LSE cancellation is shown in red. It is so highly degraded that its value is close to 0.75 for all C-SNRs.

Notice that the SER without radar interference cancellation is highly degraded (~0.75) where increases in C-SNR do not result in any improvement [1]. In other words, increasing C-SNR does not mitigate radar interference without any cancellation algorithm. From Figure 6, it is clear that long collection times result asymptotically in ideal SER since a very large N results in a very accurate radar signal estimate with small variance, no matter how small or large the QPSK energy is, as indicated in Equation (26).

Next, we consider the short collection time (or real-time case) where N is not large. Indeed, such a case is interesting because the estimate may not be as good and the variance is increased compared to the case of large N. In our Monte Carlo experiments,

we utilize N = 8, 16, 32 and 64 given that the radar-to-communications power ratio (RCR) is 20 dB. The corresponding SERs are shown in Figure 7. Again, these results only account for the fixed-phase case [1]. The variance shown on the upper x-axis is for N = 8.

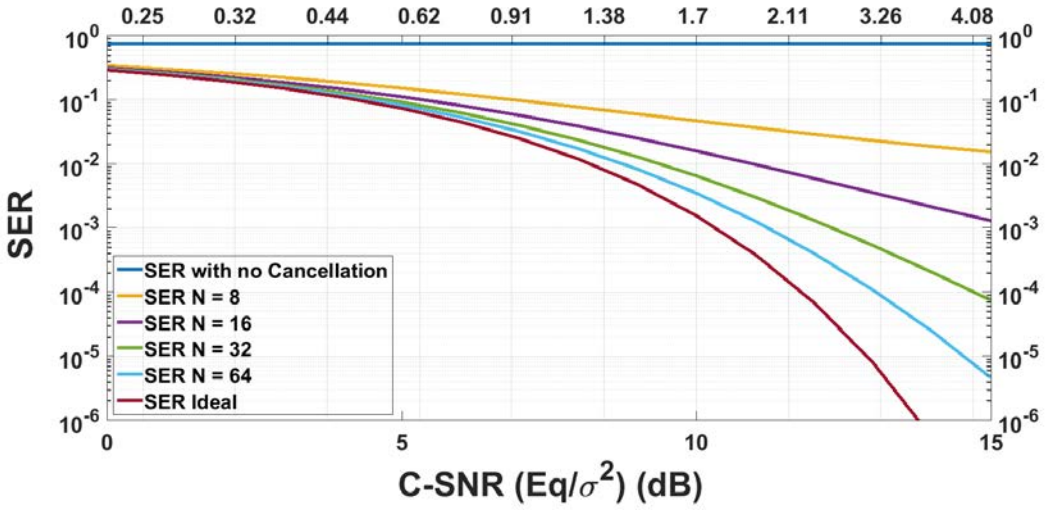

Figure 7. SER Performance as a Function of Signal Collection Time (N), LSE Variance (top Axis) and Communications-to-Noise Ratio (C-SNR) (bottom Axis). Adapted from [1].

The ideal SER (no interference), the SER with interference cancellation for various values of N and the SER without interference cancellation are shown. We again note that without interference cancellation the SER is approximately 0.75 for all values of C-SNR. Since the variance of the radar estimate is a function of increasing QPSK signal energy and noise as evidenced in Equation (26), we note the variances displayed on top of the SER figure (top of x-axis) to illustrate that the estimation variance also affects the SER. [1]. We note that number of samples taken plays a dramatic role in driving the performance curve of the signal cancellation technique toward the ideal SER. The SER performance of the cancellation technique improves as the number of samples N increases. This is expected as evidenced in the denominator of Equation (26) where large N reduces the estimation variance. It is interesting to note however that as C-SNR increases, the SER performance improves even though the variance of the estimate is

increasing. This is not necessarily an intuitive result but it does confirm the result in Equation (26) that the variance and eventually the SER are affected by the communications signal energy, noise variance and number of samples taken.

The number of samples that are used to derive an estimate are a major part of a design trade study which is heavily application dependent. Increasing C-SNR and thus the LSE variance does not have a significant negative effect on SER performance as compared to the effect of the number of samples N. If N is large enough, e.g., $N = 64$ as in Figure 7, the variance of the estimate is lowered overall, in spite of the increased variance due to the C-SNR increase.

At first glance, from Equation (13) it may seem that a large radar power would improve the estimate and the SER. As such we set up a Monte Carlo simulation where we decreased the RCR to see the effect on SER [1]. The results are shown in Figure 8. Lowering the RCR from 20 dB to −10 dB improves SER performance when no cancellation technique is utilized. The variance shown on the upper x-axis is for $N = 64$.

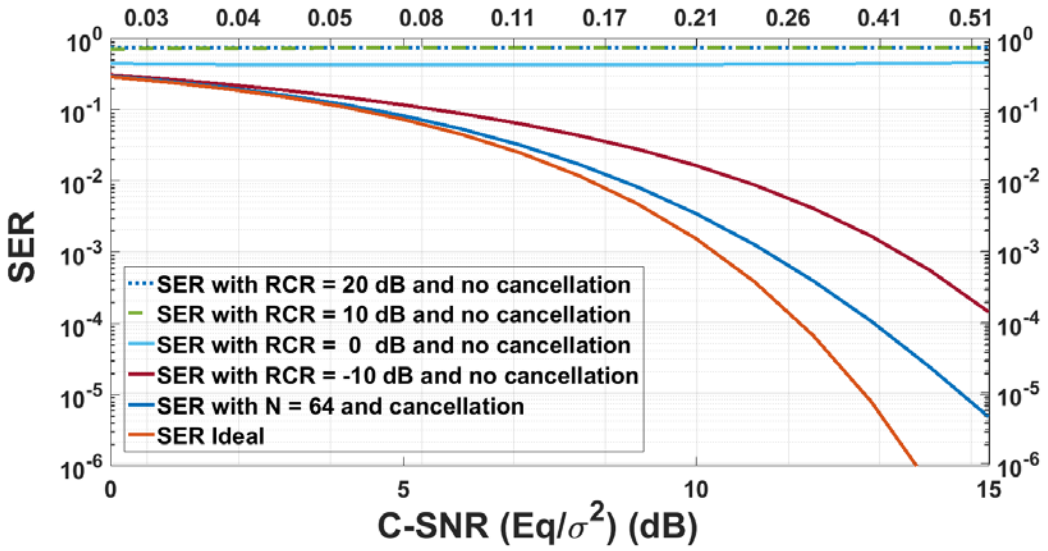

Figure 8. Varying RCR from 20 to −10 dB Shows No Change in SER Performance When LSE Cancellation Is Used. Adapted from [1].

Very interestingly, notice that the improved SER for the cancellation technique is not affected by decreasing RCR, which means that decreasing radar power has no practical effect in improving the SER [1]. As one may suspect, the 'no-cancellation' technique improves as RCR decreases but does not approach the SER of the cancellation technique, even with the RCR decreased to −10 dB.

Up to this point, the results shown have assumed the radar phase is fixed during the collection time *N*. A logical next step in the analysis is to remove the assumption that the radar phase is fixed across consecutive *N*. This leads us to investigate how many consecutive fixed-phased samples need to be taken to produce results similar to those found for a particular collection time *N*. For example, for the case of $N = 64$ in Figure 7, if the radar phase were fixed for a number of symbols less than 64, we seek to investigate the degradation in SER compared to those found in Figure 7 [1].

We set our simulation such that the fixed-phase offset is contiguous for all $N = 64$ communications symbols. This is labeled '64 Contiguous-Fixed Phases' in Figure 9, which is introduced to show the effects of varying the phase offset during the LSE sampling period. To investigate the case where a radar phase may be different during the $N = 64$ collection time, we introduce a radar phase offset different from the initial radar phase.

For the next simulation example, we set one of the 64 phases to be different. This is labeled '63 Contiguous-Fixed Phases.' In the third simulation, we set two of the 64 phases to be different. This is labeled '62 Contiguous-Fixed Phases.' For the last simulation, we set four of the 64 phases to be different. This is labeled '60 Contiguous-Fixed Phases.' Notice that the technique introduced quickly breaks down in terms of SER even when there are only two radar phase offsets that are different from the initial fixed-phase offset. This is because the LSE technique assumes that phase is constant for all collection time *N*. In other words, the technique has to be modified to account for varying phase [1]. The various contiguous-fixed phase samples are 64, 63, 62, and 60, respectively. The variance shown on the upper *x*-axis is for $N = 64$.

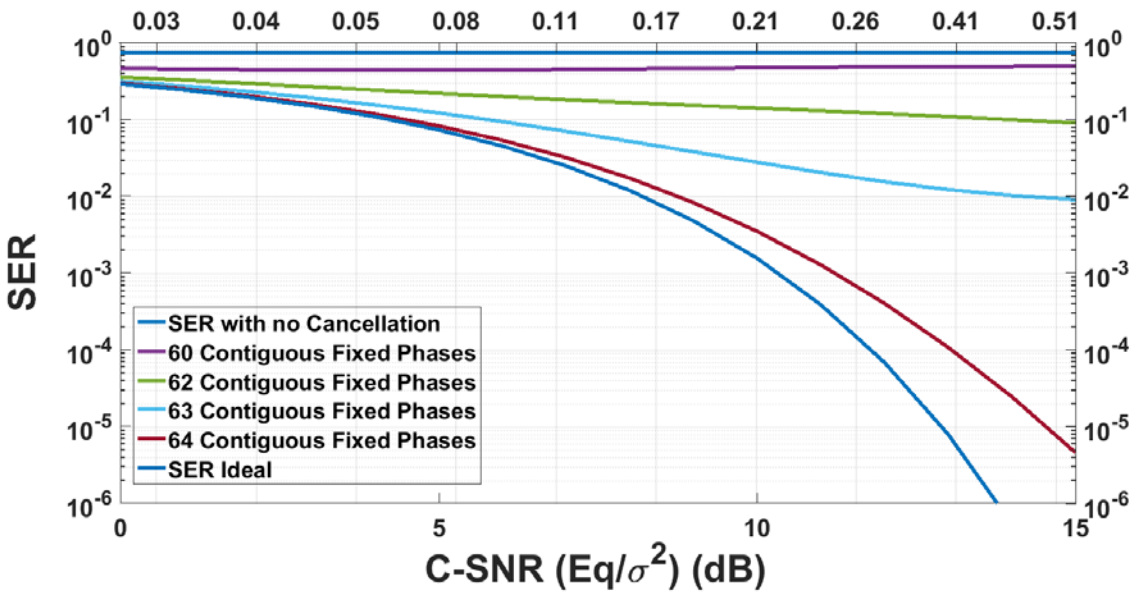

Figure 9. The "Contiguous-Fixed Phases" Curves in Each Plot Is the Number of Consecutive Radar Pulses for which the Phase Is Fixed within a 64-Sample Collection Period. Adapted from [1].

2. Results for Phase-Sequence Case

Finally, we show the results in which we revise the constant phase assumption to assume that the radar phase changes as dictated by a sequence. The process described in Figure 3. The alignment of the received signal with the phase estimate to illustrate subtraction and cancellation is shown in Figure 4. The corresponding SER results are shown in Figure 10. The variance shown on the upper x-axis is for $N = 64$.

The SER with $N = 64$ is shown to be nearly identical for the known-phase sequence when compared with the fixed-phase assumptions. Again, the phase offset is fixed and is estimated through the LSE. The phase-sequence component of the interference is assumed to be known. For our results utilizing the phase sequence, we set the sequence to be 64 samples long.

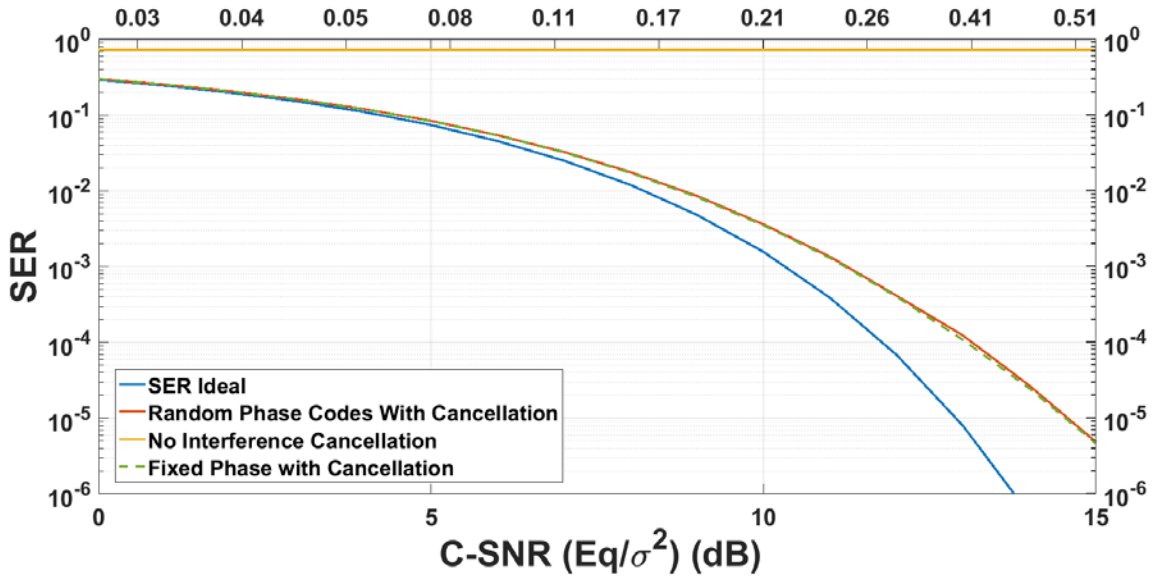

Figure 10. SER Performance of the Phase-Sequence and Fixed-Phase Assumptions Showing Nearly Identical SER Performance

As mentioned previously, we assume that the receiver does not miss the any portion of the transmitted signal. This assumption is practical since the RCR is large. This allows us to subtract the known-phase sequence as shown in Figure 11. The cancellation process (subtraction) produces the fixed-phase offset for each sample.

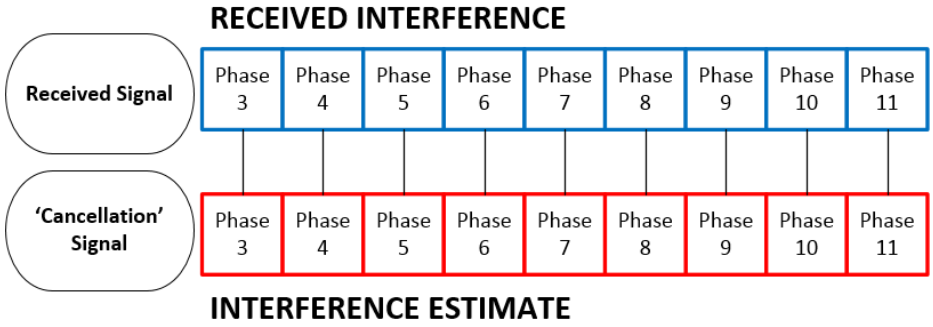

Figure 11. The Received Interference Phase Cycle Estimate Is Aligned with the Corresponding Phase in the Interference Signal Sequence

THIS PAGE INTENTIONALLY LEFT BLANK

III. FPGA IMPLEMENTATION

We have shown it is theoretically possible to demodulate the QPSK signal in the presence of high power radar interference given some of the assumptions described in Chapter II. In this chapter, we describe the design and instantiation of our signal model in hardware and compare the results with those from the previous chapter.

The workflow which extends from the signals modeled in MATLAB to the hardware implementation discussed in this chapter is referred to as model-based design. Each step of our model-based design is identified in Figure 12 and is explained individually in this chapter. The intent is to compare SER performance in hardware to that of the signal model using the hardware design process and tools described in the Appendix. A further intention is to compare QPSK demodulation performance between the initial Simulink system data types and those data types found in FPGAs, namely fixed point. The initial system is instantiated in floating-point data types versus the fixed-point data types found in Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL). Lastly, we compare the performance in VHDL synthesized in hardware versus those of fixed-point simulations.

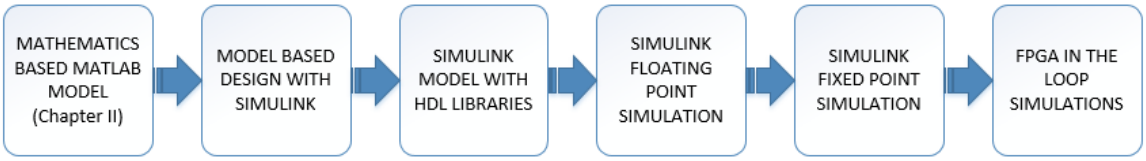

Figure 12. Model-Based Design Process Starting from the Signal Model and Progressing to FPGA in the Loop Simulations

This thesis is part of an ongoing project within the school, and, as such, there are some details of hardware performance and selection that are not pertinent to the work in this thesis. We include a more thorough discussion of the hardware development kit details and design tools in the Appendix for reference.

A. SIMULINK SYSTEM

There are two subsystems that constitute the Simulink system shown in Figure 13 The 'Computer Based' subsystem in red is composed of those functions which are not instantiated in VHDL on the FPGA. The second section in blue, 'FPGA Based,' is composed of those functions which are instantiated on the FPGA in VHDL.

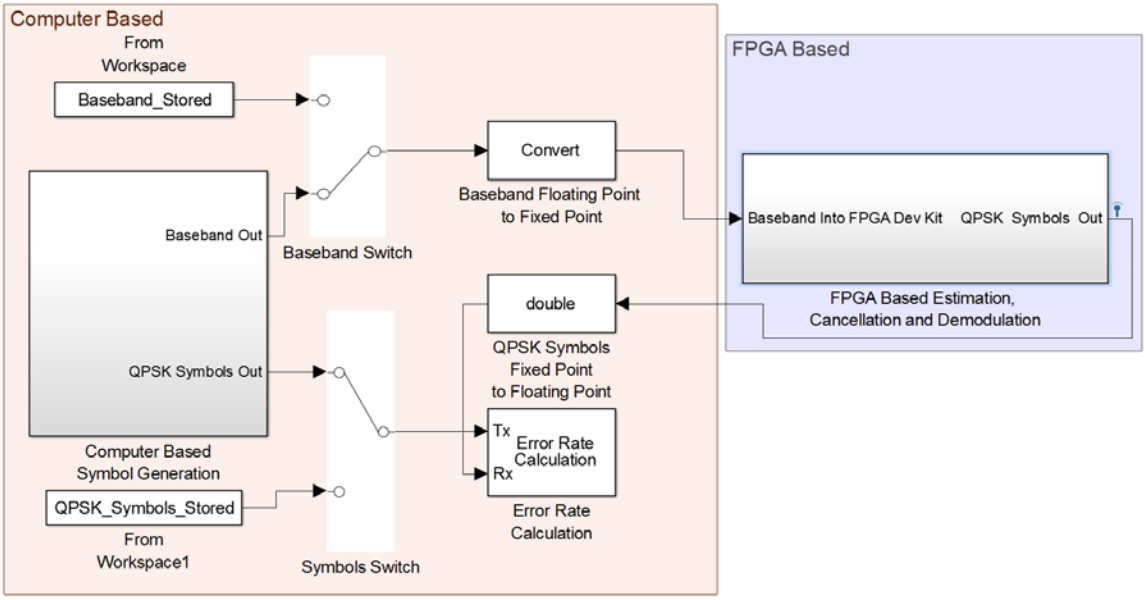

Figure 13. Top-level View of the Simulink System Showing the 'Computer Based' and 'FPGA Based' Subsystems

The design of the system in Figure 13 is the second phase of the model-based design process depicted in Figure 12. The third phase shown in Figure 12 is the verification of the 'FPGA Based' subsystem as being designed with blocks from the 'Fixed-Point Designer' toolbox, which are capable of being converted to VHDL. See the Appendix for more details on this step of the design process. The fourth phase shown in Figure 12 executes the system using double precision floating-point data types. The fifth phase shown in Figure 12 implements fixed-point data types in the 'FPGA Based' subsystem and simulates the system. The results from the floating-point and fixed-point implementations are compared for QPSK symbol recovery performance comparison. The

last phase of the design process shown in Figure 12 generates VHDL from the 'FPGA Based' subsystem and uses it to program the FPGA development kit.

Once the FPGA is programmed, it runs in parallel with the Simulink system using the fixed-point implementation. The symbol recovery performance of the hardware is compared to the fixed-point results. This process allows results to be generated at every step of the design process. We compare the results at the last step shown in Figure 12 to the results of the floating-point simulations. Further results are compared to the results from Chapter II through a SER plot. A comparison between the results obtained for this chapter and the results of Chapter II is made to demonstrate the similarities and differences between the Simulink system results and those of the signal-model algorithms.

1. Computer Based Signal Generation

The 'Computer Based' subsystem in Figure 13 generates the baseband radar interference, communications and noise signals. The QPSK symbols and baseband noise and communications signals this subsystem generates are random and saved to the workspace of MATLAB.

This random data is saved to the MATLAB workspace in order that it may be input into any given simulation by utilizing the switches in Figure 13. When comparing symbol recovery results between fixed and floating-point data type implementations the input baseband signal and QPSK symbols must be equivalent. This does not negate the fact that the communications signal and noise loaded from the workspace are still random and assumed to be independent. The switches in Figure 13 are used to load the saved baseband signals and QPSK symbols into any simulation. Reusing this random input data throughout the process allows for comparison of QPSK symbol recovery performance at each step of the conversion process.

a. QPSK Generation

The signal generation blocks which remain on the computer for all simulations are shown in Figure 14, which constitutes the subsystem ‘Computer Based’ subsystem in Figure 13.

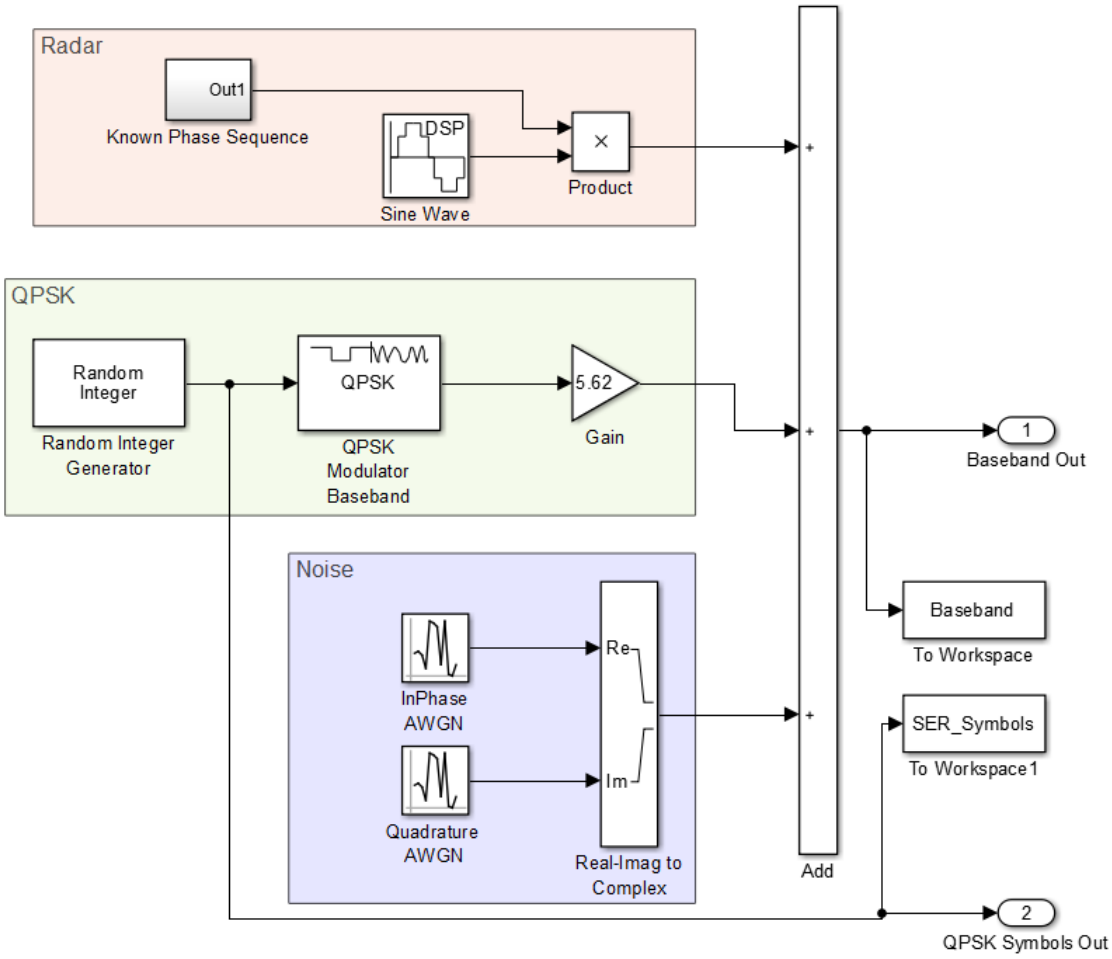

Figure 14. The Simulink System Generation of Radar, QPSK and AWGN Noise Signals

The first block in the QPSK signal generation path generates a stream of random integers from the set [0,1,2,3]. The block definitions used for the ‘Random Integer Generation’ block in Simulink are shown in Figure 15.

The ‘Random Integer’ block feeds the ‘QPSK Modulator Baseband’ block. This second block takes each of the values from the set defined in Figure 15 and maps them according to a user-defined constellation.

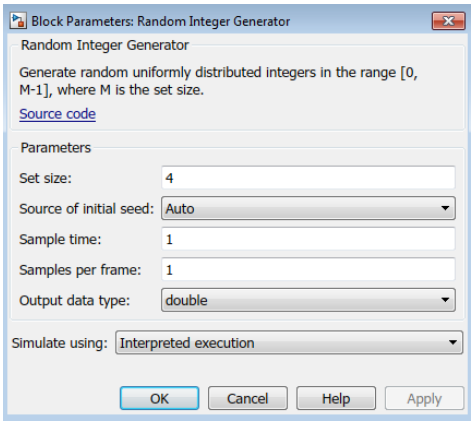

Figure 15. Random Number Generator Definition in Simulink as an Input to the QPSK Baseband Generator

The definition used in our system is shown in Figure 16. The phase offset is defined as π/4 radians. The output of the ‘QPSK Modulator Baseband’ block has an amplitude of 1.0.

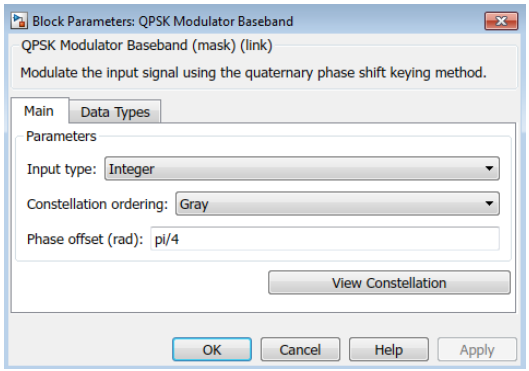

Figure 16. QPSK Baseband Modulator Definition in Simulink in terms of Gray Constellation Offset at π/4 Radians

The constellation ordering is defined as Gray. The definition of a Gray ordering with π/4 offset is depicted in Figure 17.

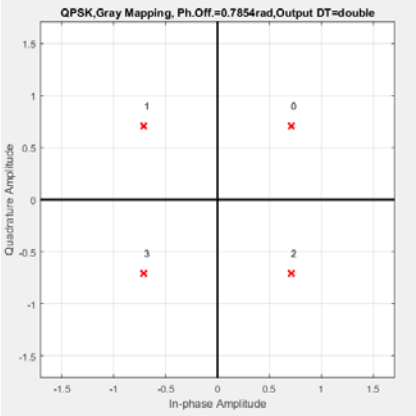

Figure 17. QPSK Baseband Gray Mapping Constellation. Symbols 0, 1, 2 and 3 are Displayed in Their Respective Quadrants

The QPSK communications signal is amplified after the 'QPSK Modulator Baseband' block by a gain function as shown in Figure 14. This gain function allows for defining the various C-SNR values used in the MATLAB simulations to generate the SER results as defined in Table 1.

Table 1. Definitions of Radar, Communications and Noise Signal Amplitudes and the Corresponding Power Levels

Amplitude			Power			Ratios	
Noise	QPSK	Radar	Noise	QPSK	Radar	C-SNR	RCR
1	1.43	14.3	0 dB	3 dB	23 dB	3 dB	23 dB
1	1.78	17.8	0 dB	5 dB	25 dB	5 dB	20 dB
1	2.24	22.4	0 dB	7 dB	27 dB	7 dB	20 dB
1	3.16	31.6	0 dB	10 dB	30 dB	10 dB	20 dB
1	3.55	35.5	0 dB	11 dB	31 dB	11 dB	20 dB
1	3.98	39.8	0 dB	12 dB	32 dB	12 dB	20 dB
1	4.47	44.7	0 dB	13 dB	33 dB	13 dB	20 dB
1	5.01	50.1	0 dB	14 dB	34 dB	14 dB	20 dB
1	5.62	56.2	0 dB	15 dB	35 dB	15 dB	20 dB

b. Radar Generation

Creation of the radar interference is accomplished by multiplying the phase-sequence block with a fixed-phase radar signal with constant amplitude. This 'Radar' block allows the RCR to be set by allowing an amplitude as well as phase to be parameterized, as shown in Figure 18.

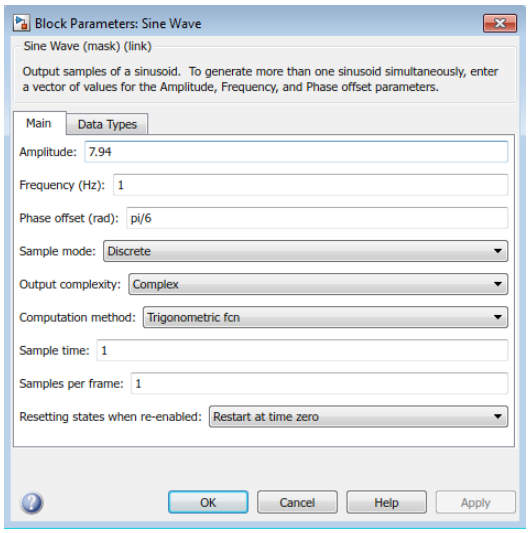

Figure 18. Constant Phase Radar Amplitude and Phase as Defined in Simulink and Output as a Complex Value

The multiplication of the phase sequence and fixed-phase offset is shown in Figure 14. The fixed-phase offset used is $\pi/6$. The 'Phase Sequence' block utilizes the same sequence as used in the signal model of Chapter II.

c. Noise Generation

The baseband noise is parameterized as shown in Figure 19 with zero mean and a variance of 1.0. As with the signal model, the variance of the noise is chosen to remain at 1.0. The interference radar and communications signal amplitudes are varied to obtain the desired RCRs and C-SNRs for each *x*-axis point along the SER curves. These values are listed in Table 1. The noise in-phase and quadrature values are generated separately and then combined into a complex value before being added to the complex communications and radar signals, as shown in Figure 14.

We sum the radar, communications and noise signals as shown in Figure 14 and route them to the MATLAB workspace and to the 'FPGA Based' subsystem that is eventually instantiated on the FPGA. The respective C-SNRs and RCRs that are shown in the ratio columns of Table 1 are the values are used to obtain the hardware co-simulation SER results given at the end of this chapter.

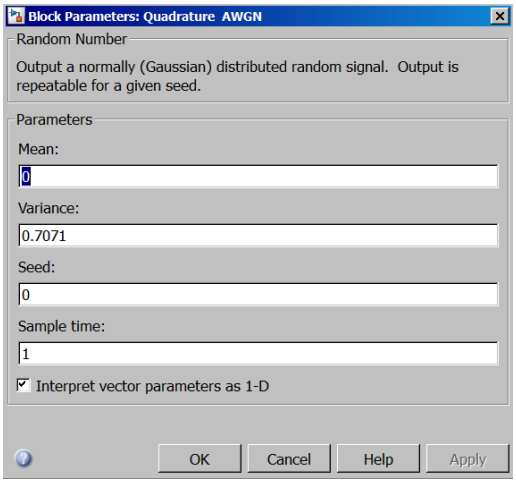

Figure 19. Noise Definitions for Both the In-phase and Quadrature Components of the AWGN Noise Source

d. Error Rate Calculation Block

The 'Error Rate Calculation' block in Figure 13 compares symbols from the estimation, cancellation and demodulation process in the 'FPGA Based' block with those that originated in the 'Computer Based' block. The 'Error Rate Calculation' block is a library block in Simulink that compares two inputs and generates three outputs as a 1 x 3 array in MATLAB. The two inputs are two different arrays of data of equal length which are compared cell by cell. For our system, the two arrays to be compared are the symbols sent from the 'Computer Based' block and the symbols demodulated by the 'FPGA Based Cancellation and Demodulation' block; thus, the SERs are calculated by this block. The three values in the output array are SER, number of errors that occurred between each of the two inputs, and the number of cells compared between the two

inputs. The first output of this array is the SER, which we use to produce the SER plots for each C-SNR and RCR. A value of $N = 10^7$ is used.

2. FPGA Based Estimator and Demodulator

The 'FPGA Based Cancellation and Demodulation' subsystem contains the angle and magnitude estimation subsystem (blue), the cancellation function (green), and finally, the QPSK demodulation function (red), respectively, in Figure 20. The description of these three subsystems follows.

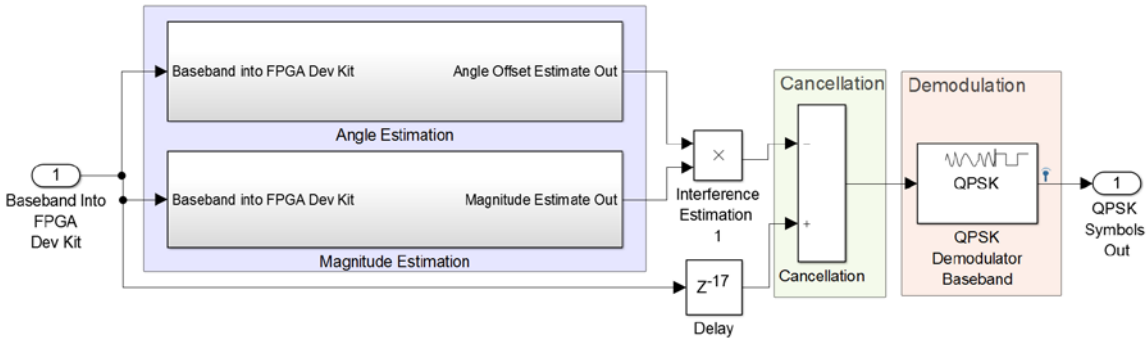

Figure 20. Four Main Subsystems to be Instantiated in the FPGA Development Kit in VHDL

a. Phase Estimation

The first step in the phase offset estimation process converts each complex sample to an angle value between 0 and 2π. This occurs in the blue section of Figure 21. We assume that the first sample received is also the first sample of the interferences phase sequence. Next, we subtract the first phase value in the sequence (green block) from the first sample received. We assume the received signal to start at the beginning of the phase sequence as we did in Chapter II; thus, this subtraction extracts the phase offset for each sample. This occurs in the 'Add' block between the blue and purple sections of Figure 21. Note: this 'Add' block is configured to subtract the two inputs, the received phase sample and the known-phase sequence.

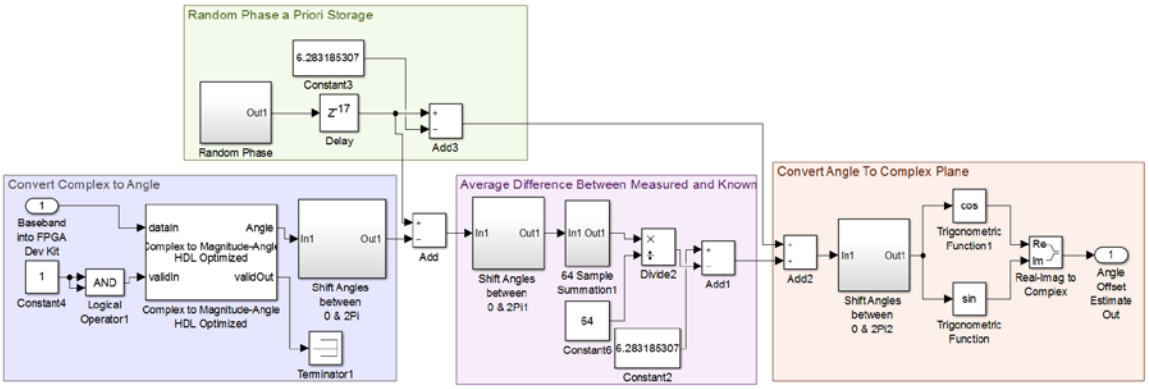

Figure 21. Phase-Sequence and Fixed-Phase Estimation System of the 'Angle Estimation' Subsystem in Figure 20

The purple section of Figure 21 is where *N* samples that result from the preceding subtraction are summed, then divided by *N*. Again, this is our LSE process. Finally, the phase-sequence components defined in Equation (5) are added to the fixed-phase offset from Equation (5). This occurs in the red portion of Figure 21. We now have an estimate of the offset. Lastly, we add the known-phase sequence for a full estimate of the phase of the radar interference. The magnitude is estimated next.

b. Magnitude Estimation

The other parameter estimated to complete our interference estimation is the magnitude, which is very straightforward and displayed in Figure 22.

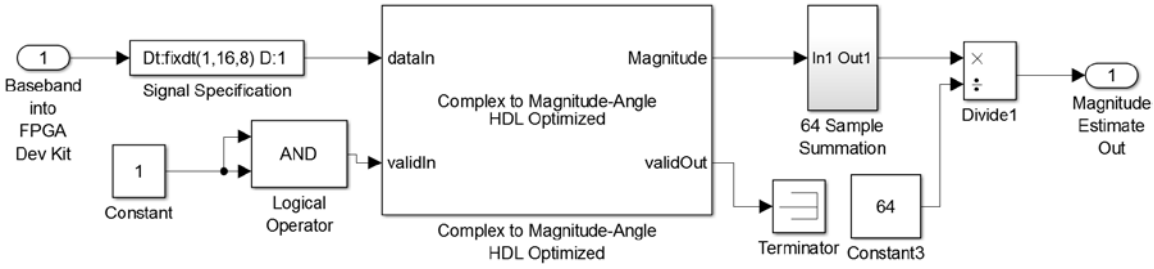

Figure 22. The Magnitude Estimation Process through Built in Simulink Block 'Complex to Magnitude-Angle HDL Optimized'

The ‘Complex to Magnitude-Angle HDL Optimized’ block yields a magnitude sample. We again sum 64 consecutive samples in the ’64 Sample Summation’ block shown in Figure 22. We divide the summation by 64 and output the magnitude estimate to be multiplied with the angle estimation.

The ‘64 Sample Summation’ subsystem whose constituent blocks form a 64 sample tapped delay line is shown in Figure 23. With the magnitude and fixed-phase estimation complete and knowing the phase sequence, we produce an estimation to be used for cancellation. This process is shown in Figure 20 as the multiplicaiton of the magnitude and phase estimation section outputs.

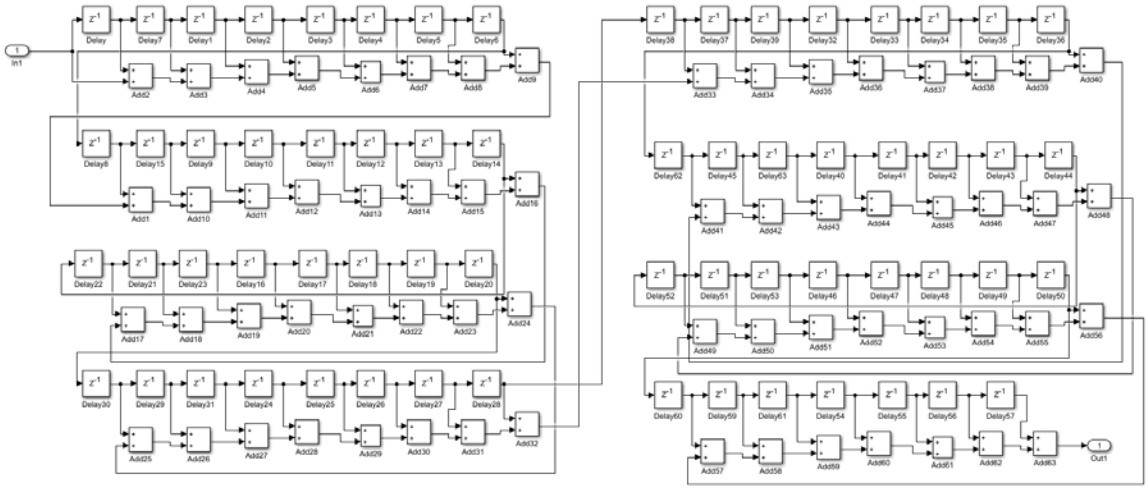

Figure 23. Sixty-Four Sample Delay-Line Adder

It is very important to note that the estimation process happens in parallel with the cancellation and demodulation process. This was not the case in Chapter II. In Chapter II the estimation process worked in series with the cancellation and demodulation procedures. In other words, the first ten interference samples that enter the hardware system are estimated with only $N = 10$ samples even though the estimator is designed for $N = 64$. Only when 64 samples have entered the system will the full performance of the estimator be reached; thus, the first 64 interference samples are not estimated, subtracted

and demodulated with the full performance of the LSE process in the hardware implementation.

c. Cancellation

With the phase offset and magnitude estimates calculated, we subtract the estimate from the received signal. This is the green area in Figure 20. The output of this process is passed to the demodulation subsystem.

d. Demodulation

The QPSK demodulation process is very simple due to Simulink having a QPSK demodulator block included in its HDL based library. The cancellation result is fed from the subtraction block output to the QPSK demodulator input as shown in Figure 20. The output of this QPSK Demodulator block is sent to the 'Error Detection Block' to compare each demodulated symbol with the symbol originally generated on the computer system. This comparison allows for generation of SER plots using MC methods.

B. FIXED POINT DATA TYPES AND VHDL CREATION

The Simulink system at this point is designed using floating point data types. This is common for simulations that remain in computer software; however, this LSE algorithm and associated signal processing is to be implemented in hardware. The process of porting the 'FPGA Based Estimation, Cancellation and Demodulation' block in Figure 13 to reside on the FPGA entails converting this block to HDL. HDL is implemented with fixed-point data types as opposed to floating point.

The work flow process that entails converting the floating-point implementation to fixed-point through the MathWorks and Xilinx design tools is discussed in the Appendix. The performance difference between the floating and fixed-point data types in terms of symbol demodulation is discussed in Section C. It is important to recall that in order to compare the demodulation results, both implementations must be presented with the same data and symbols.

The last step in the model-based design process entails converting the fixed-point implementation to an FPGA programming file. This is an automated process within the design tools discussed in Appendices A and B. The output of this process creates a 'bit' file which is used to program the FPGA with VHDL. The hardware co-simulation process is shown in Figure 24.

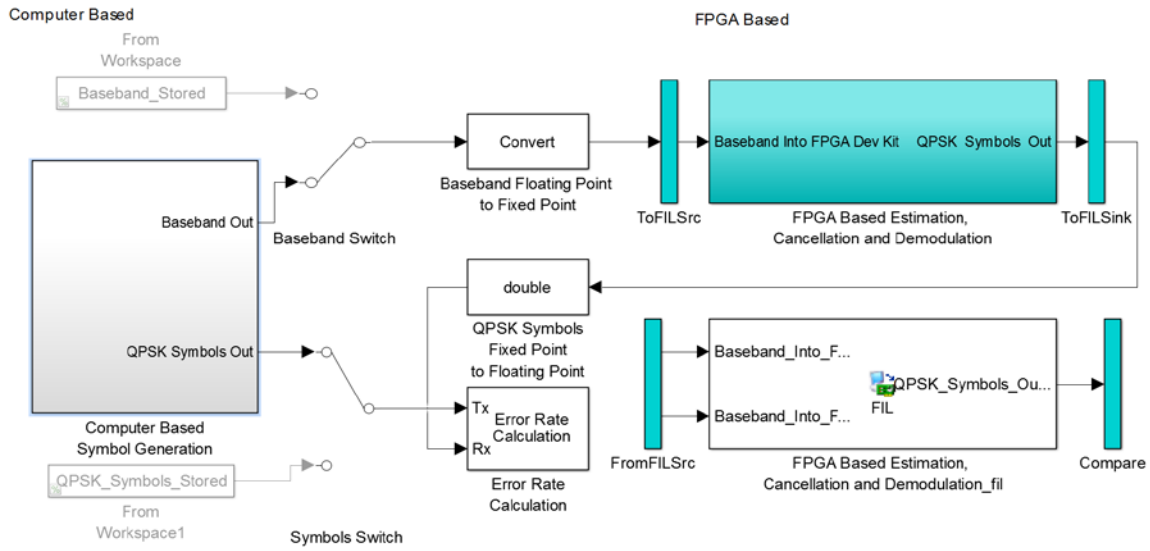

Figure 24. The 'FIL' Block Is Used to Program the FPGA on the Development Kit

The 'FPGA Based Estimation, Cancellation and Demodulation' block in Figure 24 is a Simulink fixed-point implementation run on the computer through the 'ToFILSrc' port in blue. The same data is sent through the 'FromFILSrc' port in blue to the FPGA on the development kit through an Ethernet cable between computer and development kit. This simulation allows for QPSK symbol demodulation performance comparisons between the FPGA and the fixed-point implementation of the same block in Simulink on the computer. These results are shown in the next section.

C. SIMULINK AND HARDWARE CO-SIMULATION RESULTS

We started the design in this chapter by creating a Simulink system divided into subsystems meant for FPGA implementation and subsystems meant to remain in

Simulink (software). The first simulations run are the floating-point and fixed-point implementations of the subsystem to be implemented on the FPGA. These implementations exist in Simulink, but the fixed-point implementation is the first step in generating VHDL for the FPGA. Its performance must be compared to that of the floating-point simulations.

The demodulation performance results are shown in Figure 25. The top plot is an overlay of the demodulated fixed-point and floating-point symbols; thus, the demodulation values are either 0, 1, 2 or 3, which originated in the 'Computer Based Signal Generation' block of the system.

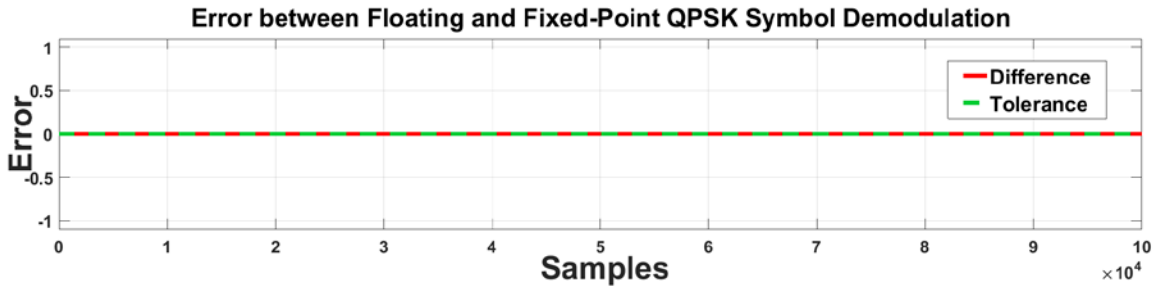

Figure 25. Results at Rx Port of 'Error Rate Calculation' Block Showing Zero Error between Floating-Point Implementation and Implementation with Fixed-Point Data Types for 10^5 Samples

Figure 25 is a plot of the difference between the demodulation of each data-type implementation on the same set of data. We see from Figure 25 that zero errors occurred. It is important to note that zero errors do not mean zero SER. What is indicated by zero error is that the fixed-point implementation is equal to the floating-point implementation in terms of demodulation performance. In other words, no demodulation performance was lost in converting from a software data type to a hardware-capable data type. This shows that any degradation of performance in terms of SER in hardware is not due to the fixed-point data type conversion. These results were generated with an RCR of 20 dB and a C-SNR of 15 dB to minimize the variance of the radar estimation and achieve high levels of cancellation.

The next set of simulation results is a comparison of the fixed-point implementation in Simulink to the actual hardware demodulation results. The comparison is made in the bottom plot of Figure 26.

The results in Figure 26 demonstrate there is again no demodulation difference between the VHDL implementation on the FPGA and the fixed-point implementation being simulated; thus, we have demonstrated that there is zero difference between the QPSK symbol demodulation performance of the hardware and both floating and fixed-point implementations of the interference mitigation algorithm in our Simulink system.

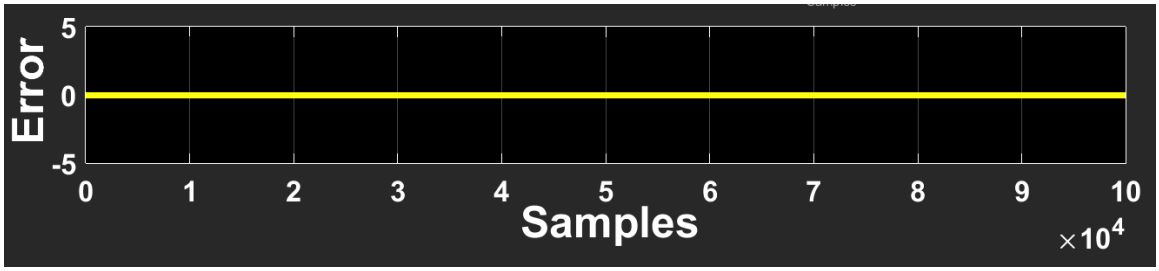

Figure 26. QPSK Demodulation Error between FPGA Co-simulation and Fixed-Point Software Simulation Showing Zero Error for 10^5 Samples

This result allows us to run simulations through the FPGA development kit and generate SER results for C-SNRs and RCRs similar to those in Figure 10 at the end of Chapter II, which displayed the results of the fixed-phase and phase-sequence assumptions.

The hardware SER performance was added to Figure 10 to create Figure 27, which depicts the difference between our theoretical software simulations and hardware SER. It is important to recall the difference is not due to the fixed-point implementations or the creation and synthesis of VHDL on the FPGA. The difference actually lies in the Simulink system itself. It is also important to recall that demodulation starts in hardware as soon as data is received. In the signal model, a full 64 samples are taken to make estimates before any of the received data is processed through the estimation, cancellation and demodulation process.

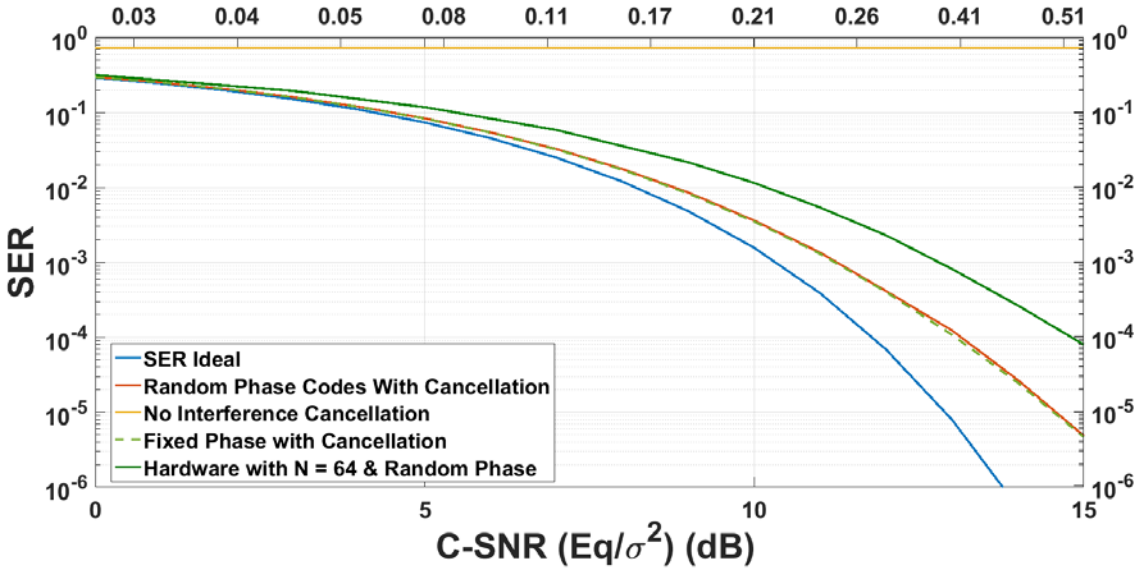

Figure 27. SER Plots of the Software Simulations with Both Fixed and Phase-Sequence Assumptions from Chapter II in Addition to the Hardware with Phase-Sequence Assumption in Chapter III

IV. CONCLUSION

This section was previously published, in part, by IEEE in [1].

In this thesis, we investigated a high-powered radar signal that interferences with the demodulation of a communications signal. We used a baseband signal model with the assumption that the signals are in the same passband channel. We assumed that the radar phase offset is both constant and is in a sequence. The phase sequence of the radar was assumed to be known; however, the radar phase offset was not assumed to be known.

To mitigate the radar interference, we proposed a form of interference cancellation where the radar signal amplitude and phase offset were estimated via LSE prior to signal subtraction. The technique produces an SER that approaches the ideal SER for very long collection times (N is very large). For lower values of N (8 to 64), we note that the resulting SER does not approach the ideal SER, but the SER greatly improves as N increases. Interestingly, the variance of the estimate increases as C-SNR increases, which in turn affects the SER somewhat negatively; however, this effect is not very pronounced when compared to the beneficial action of increasing N, since increasing N reduces the variance of the estimate to a greater degree. In other words, increasing C-SNR and, thus, the estimated variance is easily offset by increasing N [1].

A. HARDWARE IMPLEMENTATION

The same algorithms used in the signal model for estimation (LSE) and demodulation (MLD) were used in a hardware implementation. This was accomplished by building an FPGA targeted subsystem with Simulink blocks from the HDL sub-library. That subsystem was first simulated using floating-point data types. Next, the subsystem was implemented in fixed-point and then simulated. The demodulation performance was the same as the floating-point implementation for 10^5 samples. Further, HDL code was generated from the HDL Simulink blocks as well as the fixed-point definitions. The HDL code was used to program the FPGA, and a last simulation was performed with the FPGA-in-the-loop with the hardware doing the estimation, cancellation and demodulation of the QPSK communications signal. Again, the FPGA in

the loop results were the same as the floating-point simulation. Lastly, with the hardware performing as well as the fixed-point implementation, a SER plot was generated from the hardware results for comparison to the software results. The hardware results approached the software simulation results but are off by an order of magnitude by the time the C-SNR reaches 15 dB.

B. FUTURE WORK

The results obtained with the signal model show that in theory the LSE performance for the phase-sequence assumption can reach the SER performance of the fixed-phase assumption. The difference between the hardware and signal model should be investigated further since the difference lies in the Simulink system itself, not the synthesis process into hardware. The blocks in Simulink that are capable of being transitioned into VHDL are quite limited and may not allow a redesign. Performance improvements in hardware may need to be investigated within the VHDL that is generated. Additionally, allowing the hardware to build estimates before any attempt is made to estimate, cancel and demodulate any of the received signal's samples should be investigated.

APPENDIX

A. DESIGN TOOLS UTILIZED

1. FPGA Development Kit

There were a few requirements in mind when choosing the FPGA kit. This thesis is part of a larger project, and some requirements are derived from the larger project. Primarily, the FPGA development kit needs sampling speeds that are capable of sampling the highest bandwidth of expected signal. For the kit that was selected, this meant having an ADC capable of capturing assumed pulse widths as short as 50.0 ns. This meant the ADC sampling rate needed to be 20.0 MHz at a minimum. The development kit selected has a sampling rate of 250.0 MSPS for ADC and 800.0 MSPS for the digital-to-analog converter (DAC). This functionality exists within this development kit in a FPGA Mezzanine Card (FMC) [15] seen in Figure 28.

Figure 28. Kintex 7 with DSP FPGA Development Kit from AVNET. Source: [16].

Another requirement met by this development kit was having Ethernet. All three signals used in this thesis were generated on a computer rather than with test equipment such as signal generators and/or arbitrary waveform generators to generate the baseband communications, radar and noise signals. The Ethernet provides high enough data speeds

to allow for the baseband radar, QPSK and noise signals to be generated on a computer and streamed to the FPGA without a buffering process.

In addition to the Ethernet and ADC sampling rate requirements, the ADC FMC adds the benefit of making the analog signals available externally through the FMC mezzanine card as shown in Figure 29. This allows two development kits to be chained together through the FMC 150 analog ports to fully replicate the CONOP given in Figure 1, namely two systems which incorporate both a transmitting and receiving function.

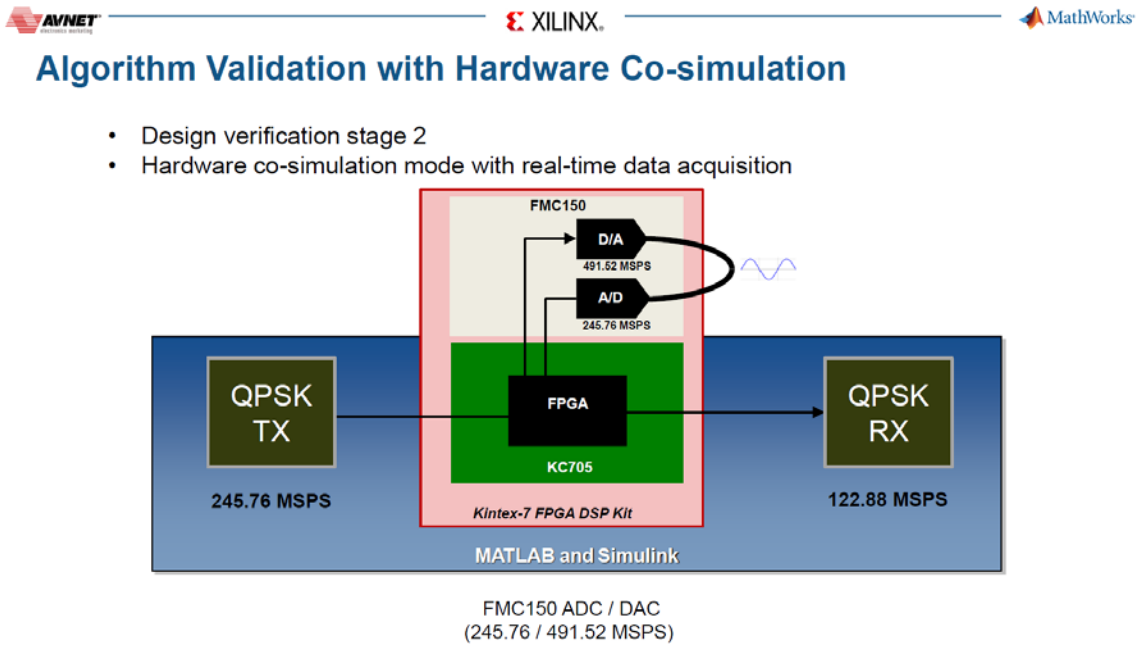

Figure 29. Functional Block Diagram of an AVNET Communications System Reference Example Also Utilizing QPSK for a Communications. Source: [17].

Lastly, this particular development kit was in a number of readily accessible examples which utilized communications systems very close to the type we implemented in hardware for this thesis; thus, this development kit alleviated the need to create new digital up-converter (DUC) and digital down-converter (DDC) designs in HDL. This allowed us to simply model everything at baseband, utilize the input and output ports in

Figure 30, and use the built in Xilinx libraries in Simulink which allowed for a model-based implementation into the FMC card.

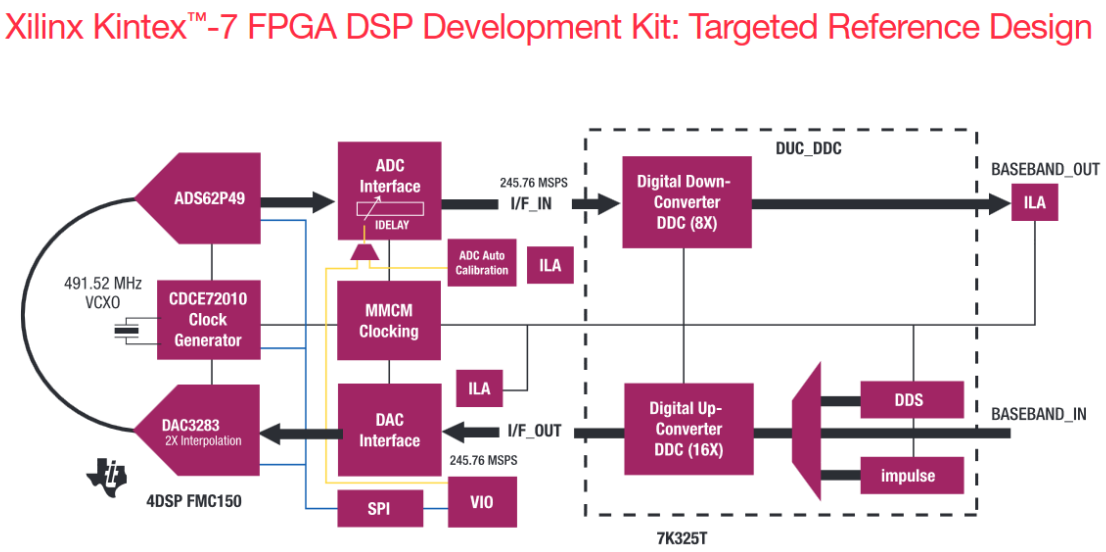

Figure 30. Function Block Diagram Demonstrating Built-in Digital Up-converter (DUC) and Digital Down-converter (DDC). Source: [16].

2. Simulink

The Mathworks produces a software package named Simulink within MATLAB which utilizes model-based design. It is referred to as model-based design due to the fact that it is based on a library of functions referred to as blocks. These blocks include a whole host of different functions, some of which can become HDL through another Mathworks tool named Fixed-point Designer. A subset of the library blocks are capable of being converted to HDL code which are target FPGA agnostic. Simulink blocks from both the HDL subset as well as blocks from the wider library were used in the Simulink system for this project. It is important to note Simulink systems must be divided at the top level into subsystems with blocks which are designated to be ported to HDL and subsystems with blocks which are not; thus, at the top level design of any given system there are one (or more) subsystem(s) which consist entirely of Simulink blocks which are from the HDL library subset.

Through the model-based design process, these subsystems get instantiated as HDL and programmed into any FPGA which can then be simulated in real time with the rest of the Simulink system still resident in software on the computer.

3. Fixed-Point Designer

While Simulink provides blocks which are portable to HDL, it assumes floating-point precision data types between the blocks in its libraries while performing simulations on the host computer. The Mathworks also produces a toolkit named Fixed-Point Designer, which aids the model-based designer in defining fixed-point data types between each respective block to implement fixed-point values throughout the subsystem dedicated for FPGA implementation. The Fixed-Point Designer aids the model-based designer to create robust and well documented HDL through a series of graphical user interface (GUI) based functional checks. These checks eliminate the need for the designer to have an understanding of the HDL language. The HDL output from Fixed-Point Designer can be created in either VHDL or Verilog standards and implemented on any FPGA. For this project we chose the VHDL standard. The Fixed-Point Designer Tool also provides real-time 'FPGA-in-the-Loop' simulations. For this thesis, that allows us to generate the baseband signals-of-interest on the computer and pass them to the FPGA for the instantiated estimation, cancellation and demodulation algorithms to be tested in real time. The results can then be compared to the full computer simulation results obtained with the system using fixed-point data types.

B. FIXED-POINT IMPLEMENTATION

The first step in using Fixed-Point Designer in the design workflow is to run an HDL compatibility report on the subsystem in the Simulink system. This process checks for, among other things, that only blocks from the HDL subset of Simulink's library have been utilized for the subsystem(s) which are to be instantiated on the target FPGA. The output of this process resembles Figure 31 in terms of no errors or warnings produced when complete.

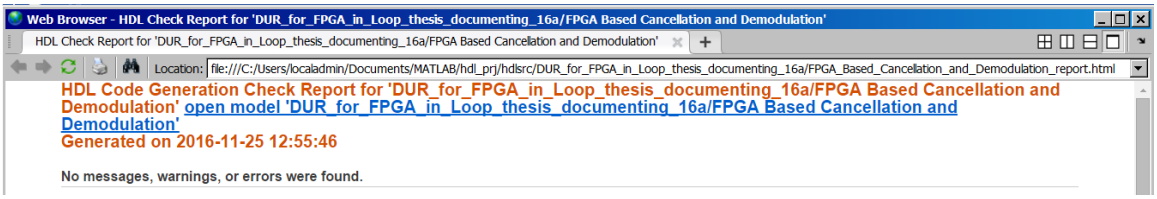

Figure 31. Check Subsystem Compatibility for HDL within the Fixed-Point Designer Toolbox in Simulink

The next step in the workflow is one of the main advantages in rapid prototyping with Fixed-Point Designer and is referred to as the Fixed-Point Tool. The first step of the Fixed-Point Tool is to use the Fixed-Point Advisor, which is shown in Figure 32. All simulations in the system used floating point data types up to this point. In order to produce efficient or even useful HDL, the subsystems that are to be implemented on the FPGA must use fixed-point data types. The Fixed-Point Tool serves two main purposes. The first is to check the systems configuration settings and prepare the relevant blocks and subsystems for fixed-point conversion. The second function prepares the system for data-type scaling. The actual process of data-type scaling is the other main advantage to rapid prototyping with Fixed-Point Designer. Once all checks have been completed, as shown in Figure 32, the system's relevant subsystems are ready for data-type scaling.

As with most of Fixed-Point Designer, the next step, referred to as data scaling, is a mostly automated process. The system is first configured using only floating-point data types. Next, the designer must determine the maximum data sizes that the prototype system will encounter. For this effort, that means determining the maximum baseband amplitudes for all three signals: noise, communications, and radar. The reason for this is to allow the Fixed-Point Advisor to automate the process of determining the fixed-point data scale for the input(s) and output(s) to all blocks within the relevant subsystem(s). The Advisor proposes all these fixed-point values based on the signal amplitudes propagating through the system. If fixed-point values are proposed using data that is lower in amplitude than what will be seen when using real data, overflow can occur due to improper selection of fixed-point values throughout the system.

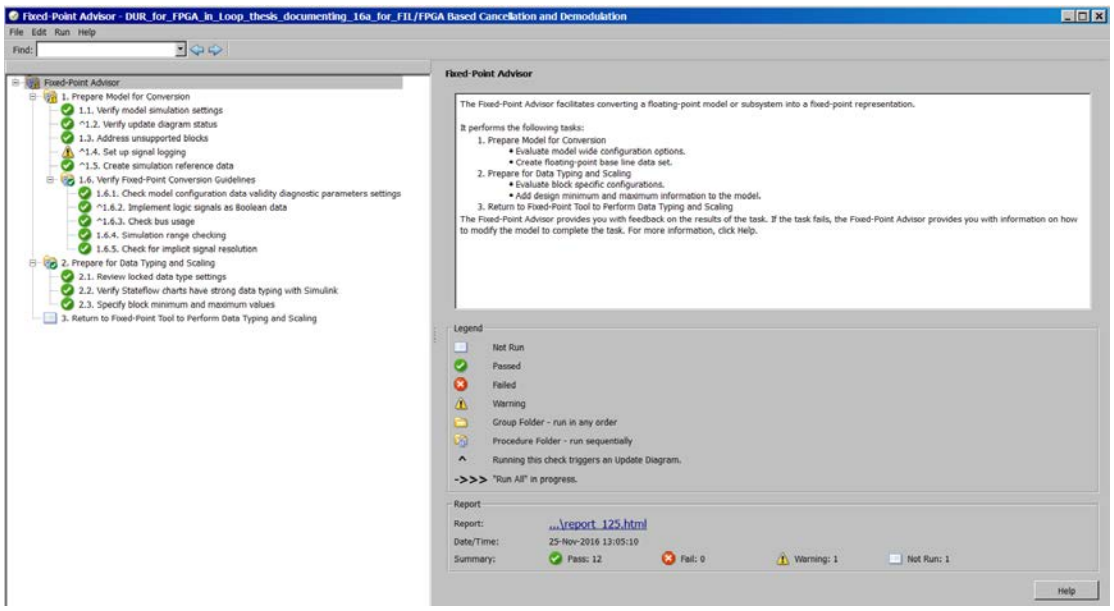

Figure 32. Fixed-Point Advisor within the Fixed-Point Tool

Once the system's data types are configured to floating-point, all three input signal amplitudes corresponding to signal power levels are chosen to represent the maximum value the actual hardware will encounter from its environment. The values that are chosen are shown in Table 1 along the bottom row.

The system is simulated using the signal power levels in Table 1 and floating-point data types. The QPSK symbols transmitted as well as the baseline signal are saved to the MATLAB workspace. Based on the results of the floating-point data type simulation, the Advisor proposes (refer to bottom of Figure 33) fixed-point data types across the entirety of the subsystem. In the case of this thesis, that is the 'FPGA Based Cancellation and Demodulation' subsystem.

The next step is to accept (refer to bottom of Figure 33) each individual fixed-point value throughout the subsystem. In this thesis, all of the data type implementations proposed by the advisor were accepted. The next step was to compare the results of the simulation between floating point and fixed-point implementations. In order to accomplish this, the input data sets must be equivalent. Since the noise and communications signal contain random data, it is necessary to store the input data from the floating-point simulation for fixed-point simulation performance comparison.

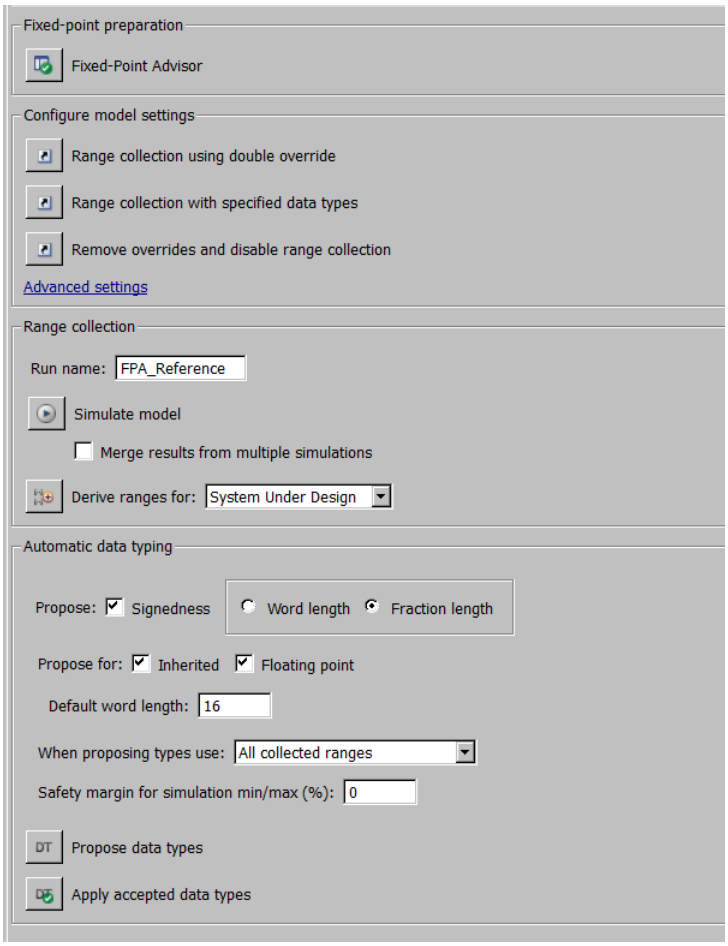

Figure 33. Fixed-Point Designer GUI for Data Type Implementation of Subsystem Designated for Fixed-Point Type Instantiation

This is accomplished by switching in the data from the MATLAB workspace and is accomplished though using the switches at the top level of the system to import data from the 'From Workspace' blocks at the system's top level as shown in Figure 13.

The floating-point values are replaced by the proposed and accepted fixed-point values throughout the subsystem. The input values are switched in from the workspace and the simulation is rerun. The results of the demodulated symbols through the 'FPGA Based Estimation and Cancellation' block are compared across the floating-point and fixed-point simulations.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] G. Meager, R.A. Romero and Z. Staples, “Estimation and cancellation of high powered radar interference for communication signal collection,” in *2016 IEEE Radar Conf.*, 2016, pp. 1–4, doi: 10.1109/RADAR.2016.7485263. © IEEE 2016.

[2] L. Cohen and R. Jost, “Spectrum analysis and measurements in a congested electromagnetic environment,” in *IEEE International Symposium on Electromagnetic Compatibility EMC,* Aug. 2013, pp. 66–71.

[3] Y. Deng, K. Jin Kim, T. Duong, M. Elkashlan, G. Karadiannidis and A. Nallanathan., “Full-duplex spectrum sharing in cooperative single carrier systems,” in *Proceedings IEEE Wireless Communications and Networking Conf.*, 2015, pp. 25–30.

[4] A. Khawar, A. Abdel-Hadi and C. Clancy, “Spectrum sharing between S-band radar and LTE cellular system: A spatial approach,” in *Proc. IEEE International Symposium on Dynamic Spectrum Access Networks,* April 2014, pp. 7–14.

[5] H. Shajaiah, A. Abdelhadi, and C. Clancy, “Spectrum sharing approach between radar and communications systems and its impact on radar’s detectable target parameters,” in *Proc. IEEE Vehicular Technology Conference*, May 2015, pp. 1–6.

[6] H. Wang, J. Johnson, C. Baker, Y. Lixin, and C. Zhang, “On spectrum sharing between communications and air traffic control radar systems,” in *Proc. IEEE Radar Conference*, May 2015, pp. 1545–1550.

[7] W. Diffie, “Chattering about SIGINT,” in *IEEE Security Privacy*, 2006, vol. 4, pp. 9.

[8] Y. Wang, D. Su and W. Chen, “Radio noise jamming analysis on communications system” in *Proc. IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, 2009, pp 854–857.

[9] H. Deng and B. Himed, “Interference mitigation processing for spectrum-sharing between radar and wireless communications systems,” in *IEEE Transactions on Aerospace and Electronic Systems*, 2013, vol. 49, pp. 1911–1919.

[10] B. D. Cordill, S. A. Seguin, and L. Cohen, “Electromagnetic interference to radar receivers due to in-band OFDM communications systems,” in *Proc. IEEE International Symposium on Electromagnetic Compatibility*, Aug. 2013. pp. 72–75.

[11] N. Nartasilpa, D. Tuninetti and N. Devroye, “On the error rate of a communications system suffering from additive radar interference,” in *Proc. IEEE Global Communications Conf.*, 2016, pp. 1–6.

[12] Destroyer Silhouette. (n.d.). clipartfest.com [Online].Available: https://img.clipartfest.com/23854c92f1bd377b30ce5c0ecc1cd15d_-resolution-3294x2028-clipart-destroyer-silhouette_3294-2028.jpeg. Accessed Feb. 19, 2017.

[13] E2 Silhouette. (n.d.). ebaymag.com [Online].Available: http://i.ebayimg.com/images/g/K-4AAOSwR0JUN28H/s-l300.jpg. Accessed Feb. 19, 2017.

[14] Kay, S. M., *Fundamentals of Statistical Signal Processing*, vol. 3, (Prentice-Hall Signal Processing Series). Upper Saddle River, NJ: Prentice-Hall, 1993.

[15] FMC150. (n.d.). 4dsp.com [Online]. Available: http://www.4dsp.com/FMC150.php. Accessed Feb. 19, 2017.

[16] Getting started guide. (n.d.). AVNET [Online]. Available: https://products.avnet.com/opasdata/d120001/medias/docus/4/k7dsp_getting_started_guide.pdf. Accessed Feb. 19, 2017.

[17] Using Xilinx System Generator for DSP with Simulink and HDL Coder. (n.d.) MathWorks [Online]. Available: https://www.mathworks.com/content/dam/mathworks/mathworks-dot-com/campaigns/products/files/using-xilinx-system-generator-for-dsp-with-simulink-and-hdl-coder.zip. Accessed Feb. 19, 2017.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California